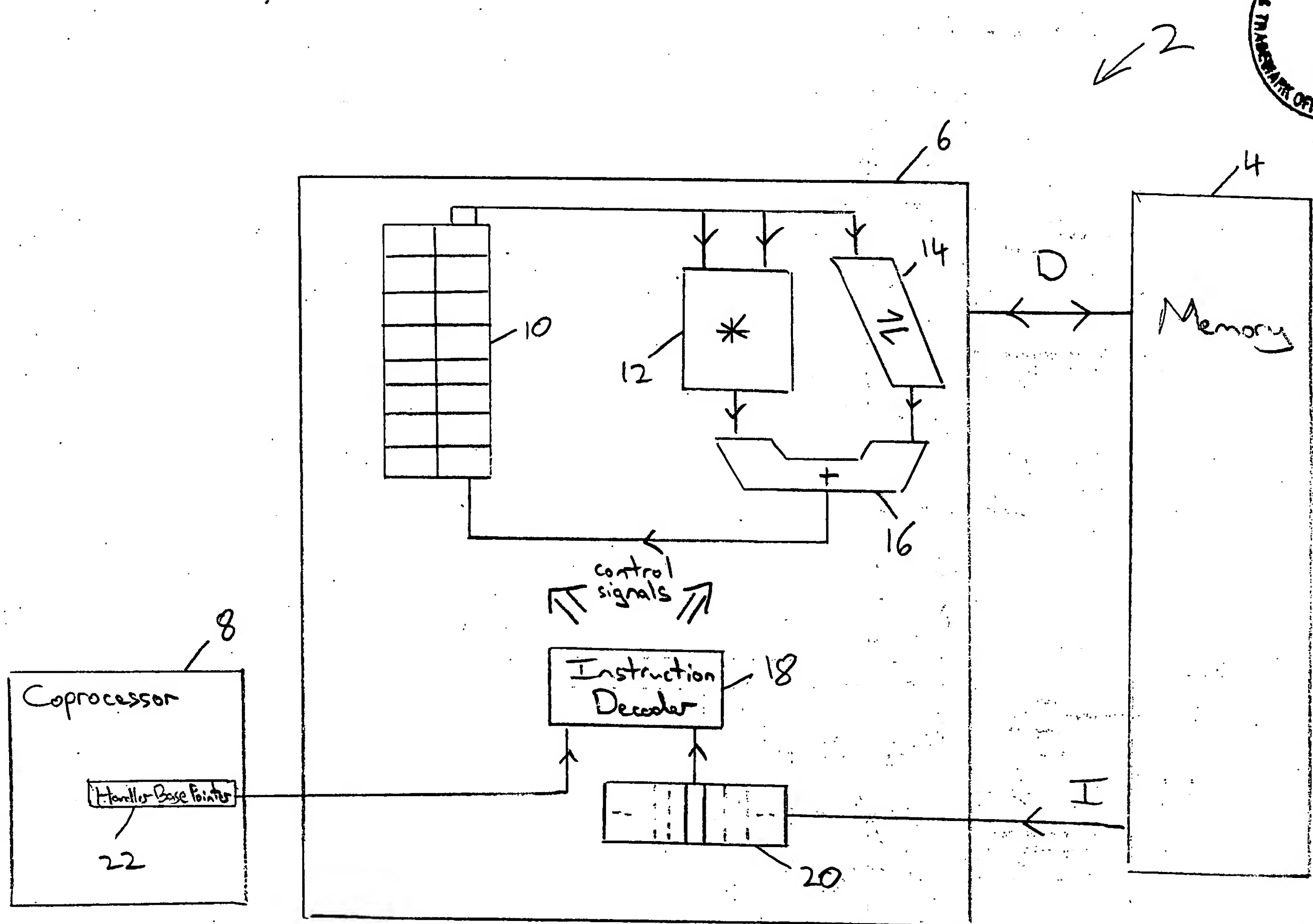

Fig. 1

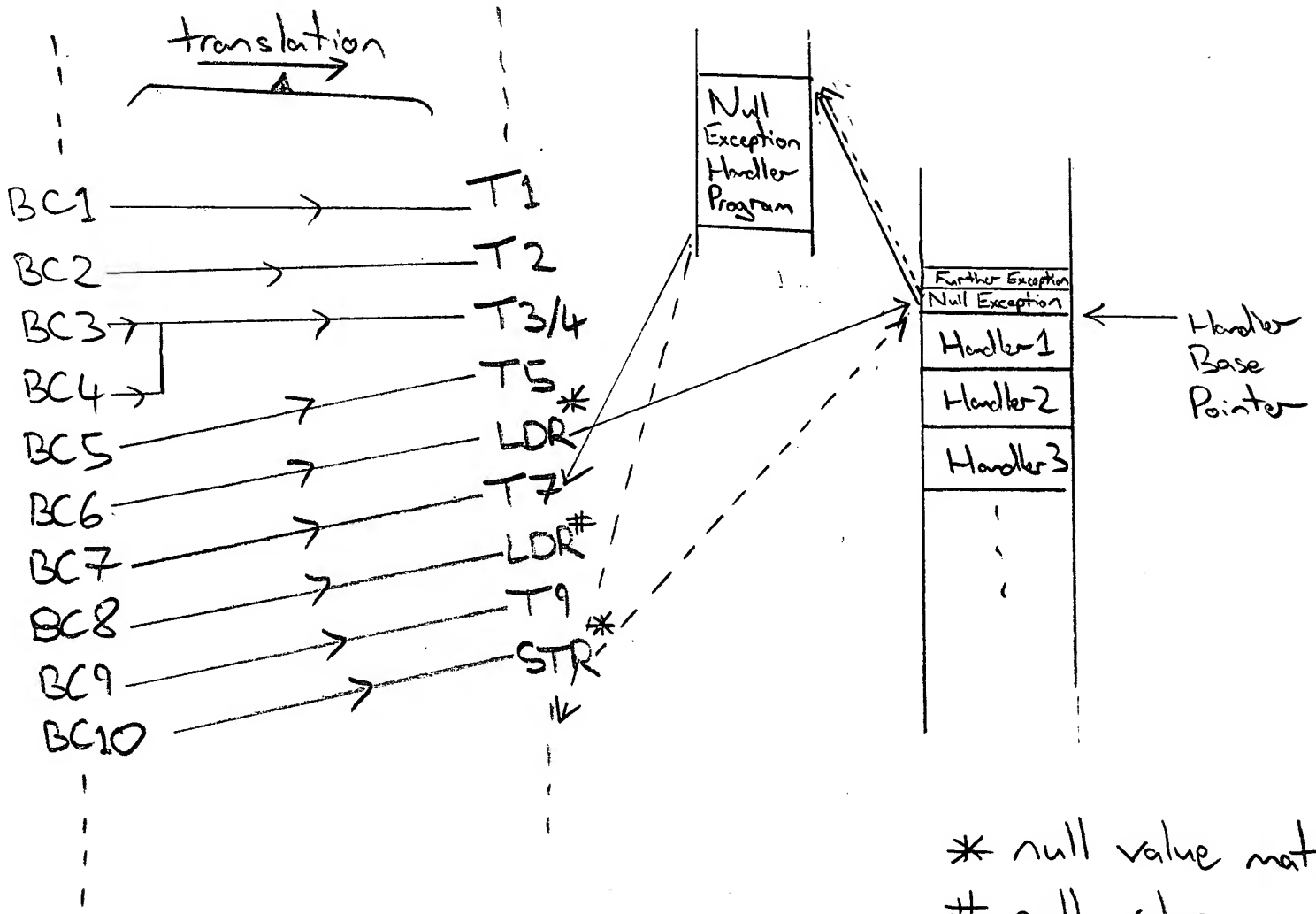
translation
BC1
BC2
BC3
BC4
BC5
BC6
BC7
BC8
BC9
BC10
T1
T2
T3/4
T5
LDR*
T7
LDR#
T9
STR*
Null Exception Handler Program
Further Exception
Null Exception
Handler-1
Handler-2
Handler-3
Handler Base Pointer
* null value match
# null value non-match
Machine Independent Instructions
Thumb-2 Instruction

Fig. 2

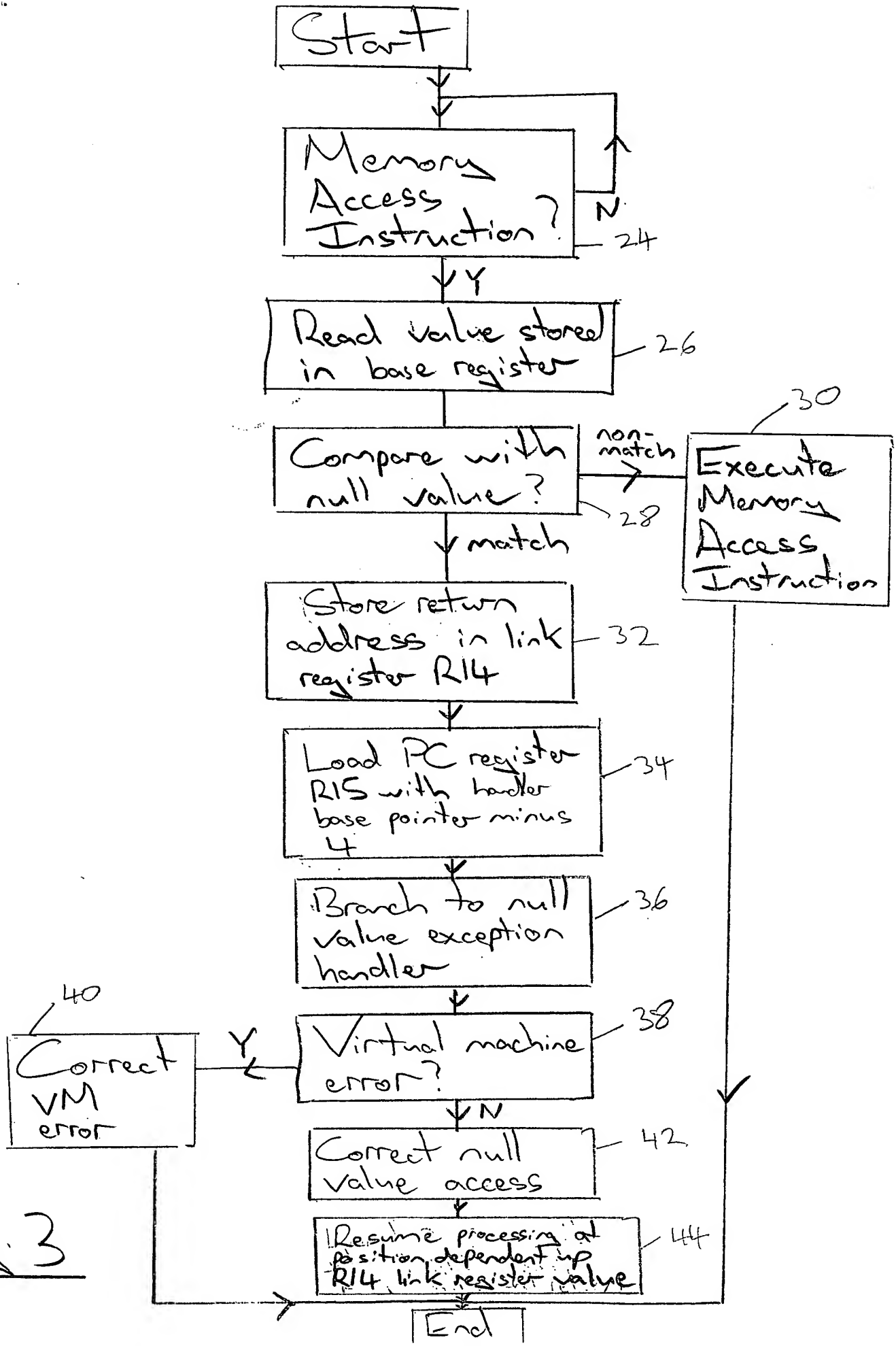
Start
Memory Access Instruction?
N
24
Y
Read value stored in base register
26
Compare with null value?
non-match
28
30
Execute Memory Access Instruction
match
Store return address in link register R14
32
Load PC register R15 with handler base pointer minus 4
34
Branch to null value exception handler
36
Virtual machine error?
38
Y
40
Correct VM error
N
Correct null value access
42
Resume processing at position dependent up R14 link register value
44
End

Fig. 3

Instruction	LDR.X Rd, [R9,#immediate.6]										
Encoding	15	14	13	12	11	10	9	8	3	2	0
	<LDR.X opcode>							#immed_6		Rd	
Thumb-2 Equivalent	LDR Rd, [R9 + #immediate LSL #2]										
Definition	Rd = [R9 + #immediate LSL #2]										

Encoding space	2^8	8 bits
Note This instruction, as are all loads and stores while in Jazelle-X state, is subject to the Null Check mechanism described in **4.3**		

Instruction	STR.X Rd, [R9,#immediate.6]										
Encoding	15	14	13	12	11	10	9	8	3	2	0
	<STR.X opcode>							#immed_6		Rd	
Thumb-2 Equivalent	STR Rd, [R9 + #immediate LSL #2]										
Definition	[R9 + #immediate LSL #2] = Rd										
Encoding space	2^8							8 bits			
Note This instruction, as are all loads and stores while in Jazelle-X state, is subject to the Null Check mechanism											

Fig. 4

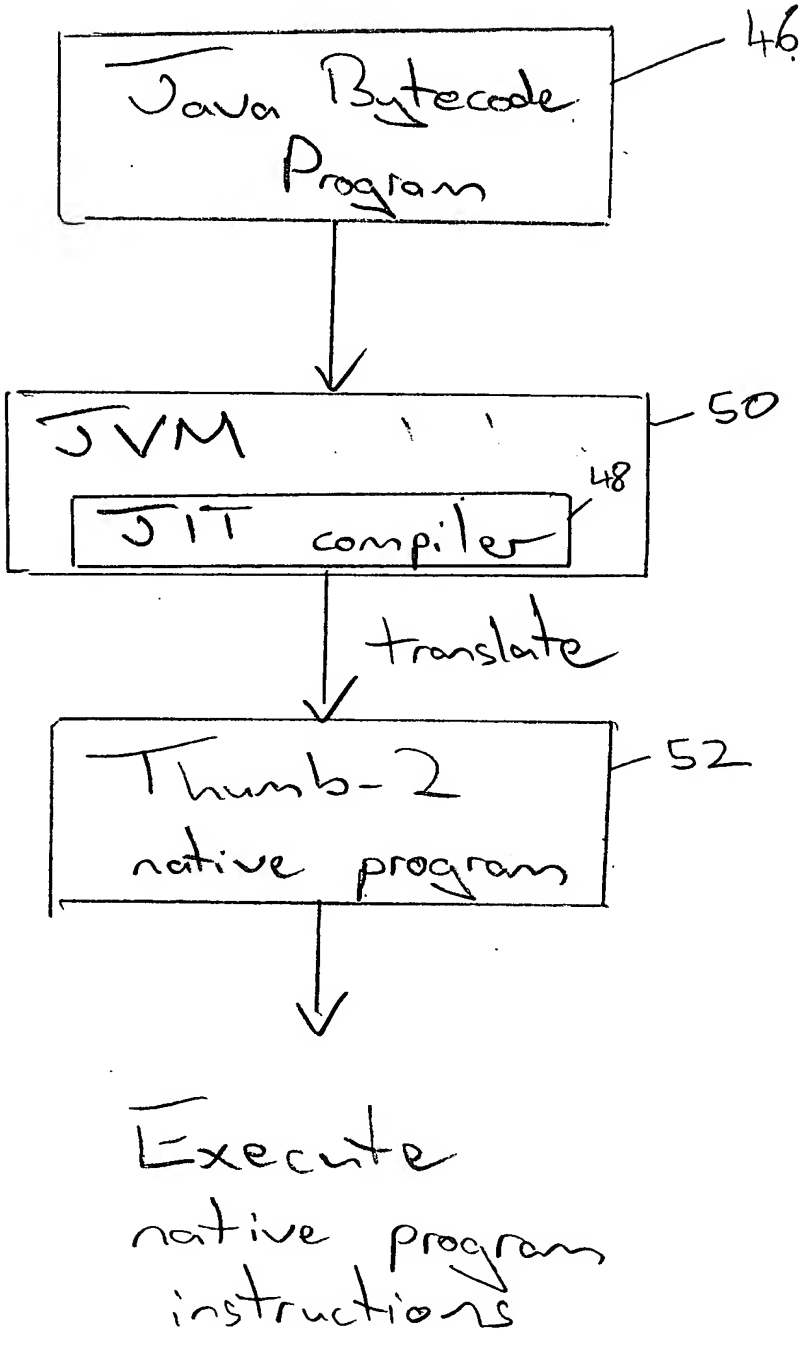
Java Bytecode Program
46
JVM
50
JIT compiler
48
translate
Thumb-2 native program
52
Execute native program instructions

Fig. 5